64 PLUS 4 & AMIGA

LUTY 1991 Indeks 377112 CENA 5000 zł

MIESIĘCZNIK UŻYTKOWNIKÓW KOMPUTERÓW COMMODORE

W NUMERZE:

- Złote TILT'Y
- Ogonki i C-64
- Operation STEALTH



W numerze :

W następnym numerze :

- **Historia komputerów**
- **Indiana Jones III Adventure**
- **Adresy firm software'owych**
- **Polskie litery na C-16 i Plus 4**

OGŁOSZENIA

ERRATA

W numerze grudniowym (1990 rok) w artykule „C-16 i okna” w listingu (na zielonym tle) brakuje ostatniej linii. Powinna ona brzmieć: **1005 RETURN.**

Na stronie 13 (oj, pech), w artykule pod tytułem „Animacji duszków ciąg dalszy” druga instrukcja w linii 10 powinna brzmieć: **P=180/PI.**

PRZEPRASZAMY

SPROSTOWANIE

Zapowiadany w styczniowym numerze artykuł o makroassemblerze C-64 zamieścimy w marcowym wydaniu naszego pisma.

Redakcja

6 4 P L U S 4

miesięcznik nr 2(4) luty 1991
cena 1 egz.: 5000 zł



Wydawca:
ABUK Spółka z o.o.
Redakcja nie ponosi odpowiedzialności za treść ogłoszeń.

Adres redakcji: Redakcja „64 plus 4” 85-166 Bydgoszcz 43 skrytka pocztowa 64
Redagują: Marcin Dudar, Sambor Kuźma, Paweł Sołtysiński, Waldemar Szczygieł (red. nacz.), Robert Turliński, Wojciech Wasilewski.
Foto: Jerzy Wiśniewski
Skład: ABUK
Druk: Prasowe Zakłady Graficzne, 85-009 Bydgoszcz, ul. Dworcowa 13
zam. 140/91

Drodzy Czytelnicy!

Składamy serdeczne podziękowania za przesłane pod naszym adresem życzenia świąteczne i noworoczne. Oby Wam i nam wszystkie one się spełniły.

Cieszy nas coraz większa ilość listów z uwagami i pomysłami dotyczącymi naszego pisma. Wszystkie je dokładnie czytamy. Na część z nich postaramy się odpowiedzieć indywidualnie, na inne odpowiedzi znajdziecie w publikowanych artykułach.

Cieszymy się również z prac napływających na nasz konkurs „Najlepszy program miesiąca". Oceniając je staramy się nagradzać autorów prezentujących ciekawe rozwiązania i pomysły. Przy okazji: jeśli w swoich pracach korzystacie z pomysłów i chwytów opracowanych przez innych autorów, to pamiętajcie o podawaniu źródeł informacji. Jest to bowiem wyraz uczciwości i szacunku dla cudzej pracy.

Zważając na ilość i jakość nadsyłanych programów postanowiliśmy przyznawać główną nagrodę co dwa miesiące. W styczniowym numerze przedstawiliśmy laureata w kategorii programów na C-64, w tym miesiącu na C-16. W marcowym numerze przedstawimy kolejnego zwycięzcę (C-64).

Zapraszamy wszystkich do nadsyłania swoich prac - jak najszybciej chcielibyśmy przyznawać naszą nagrodę co miesiąc w każdej z kategorii!

Zapowiedzieliśmy, że wśród pierwszych stu prenumeratorów naszego pisma rozosujemy nagrody. Atrakcyjne torby - dyplomatki wylosowali: Rogoża Mirosława - Wałcz, Anna Grajewska -Drzycim, Piotr Skierczyński - Olsztyn, Marek Brzostyński - Katowice, Kazimierz Mielewski - Świecie, Romuald Gracz - Poznań, Tomasz Świderski - Łódź, Mariusz Gwiazda - Biała Podlaska, Grzegorz Frost - Knurów, Tomasz Nowakowski - Tuchola. Gratulujemy! Oby były zawsze pełne dobrych programów komputerowych i pieniędzy!

Przypominamy, że aktualna cena jednego egzemplarza w prenumeracie wynosi 4600zł. Wpłat należy dokonywać na konto: Bank PKO SA Bydgoszcz, konto nr: 5.09011-400522.7-136-11-111.0. Bardzo prosimy o czytelne wypełnianie blankietów wpłat. Prenumeratę można zawierać na dowolny okres do końca 1991r.

Wszyscy, którzy wykupili prenumeratę do końca stycznia br. nie muszą dopłacać w związku z wyższą ceną pisma.

W dalszym ciągu aktualna jest nasza oferta dla chętnych do prowadzenia indywidualnego kolportażu naszego pisma (w szkole, w zakładzie pracy, w studiach komputerowych itd.). Przypominamy zasady: jednorazowy zakup nie mniej niż 20 egzemplarzy po cenie hurtowej (aktualnie 4000zł.); płatność gotówką przy odbiorze (odbiór w Bydgoszczy - możliwe inne uzgodnienia). Zainteresowanych prosimy o lisowny kontakt.

Coraz częściej napływają pytania o możliwość zakupu poprzednich numerów naszego pisma. Informujemy, że są one do nabycia w warszawskiej księgarni „Elektronika" przy ulicy Mokotowskiej 51/53 oraz za pośrednictwem redakcji (zamówione numery prześlemy za zaliczeniem pocztowym).

Redakcja

Z DALEKA I Z BLISKA ...... Z DALEKA I Z BLISKA ...... Z DALEKA I
Z DALEKA I Z BLISKA ...... Z DALEKA I Z BLISKA ...... Z DALEKA I
Z DALEKA I Z BLISKA ...... Z DALEKA I Z BLISKA ...... Z DALEKA I
Z DALEKA I Z BLISKA ...... Z DALEKA I Z BLISKA ...... Z DALEKA I

- W Londynie między 16 a 18 listopada odbył się wielki, piętnasty Commodore Christmas Show. Obecnych było ponad 100 wystawców z całego świata, rozdawano reklamówki gier, firm software'owych, można było pograć na normalnych Amigach i A-CDTV. Bilet kosztował jedyne 5 funtów. Show odbywał się na dwóch ogromnych piętrach w Novotelu w Londynie. W ramach imprezy odbyło się też sześć seminariów.

- Kolejną imprezą komputerową był Desktop Publishing Show, który odbył się między 28 a 30 listopada w Business Design Centre w Londynie, poświęcony zastosowaniom programów typu DTP.

- W dniach 6-9 grudnia na Wembley w Londynie odbył się Computer Shopper Show. Ponad 250 stanowisk wypełnionych było hardware'm, software'm, peryferiami i przede wszystkim komputerami. Wystawa pobiła wszystkie swoje poprzednie rekordy, m.in. Ilości wystawionych towarów, odwiedzających ją ludzi.

- To jeszcze nie wszystkie imprezy! W styczniu (4-6) również w Novotelu w Londynie odbył się trzeci show o nazwie „16 Bit Computer Fair” poświęcony wyłącznie Atari ST i Amigdze. Ponad 140 wystawców z całego świata prezentowało rozrywkowe i profesjonalne zastosowanie komputerów.

- Pojawiła się na rynku pierwsza całkowicie optyczna myszka. Jest bardzo ładna i kosztuje tylko 40 funtów.

- Wszyscy chyba już znają wspaniałe grafiki Tobias'a Richter'a (nasz numer jeden wśród grafików). Okazu je się jednak, że te wszystkie wspaniałe rysunki są płynnie animowane czyli tworzą film! Tobias już od dłuższego czasu zajmuje się animacją komputerową. Większa część jego prac (a przynajmniej te najlepsze) jest dostępna tylko na maszynach z minimum 1MB pamięci. Ekrany, które mieliście okazję oglądać w demie grupy Crusaders pochodzą z filmu Tobias'a pt. „Star Trek the Movie”. Niestety film ten jest dostępny tylko i wyłącznie na video! Oczywiście wszystkie ekrany i cała animacja zostały przygotowane na Amidze, lecz sam film zajmował zbyt dużo miejsca by zmieścić się nawet na rozszerzonych maszynach. Obecnie ostatnia produkcja Tobias'a (właśnie Star Trek) jest uważana za najlepszą animację wykonaną na Amidze na świecie. Mamy nadzieję, że uda nam się zdobyć ten film.

- Supra Corporation zrobiła dysk twardy do Amigi 500 o bardzo małym poborze mocy – mniejszym niż 4 waty! Na płycie może znajdować się do 8 MB pamięci. Wersja 20MB kosztuje 679 funtów.

- Uwaga, uwaga!!! Datel Electronics nareszcie wyprodukował Action Replay II. Poprawiony system operacyjny, który teraz zajmuje już 128 KB, a w nim dodatkowo: Music Sound Tracker (do wycinania muzyczek z gier i demosów), Autofire Manager (do płynnego zmieniania szybkości strzelania 0-100%), Diskcoder (do protekcji programów przed innymi użytkownikami), Preferences (do ustawiania preferencji ekranu roboczego), Disk Monitor (do zabaw z dyskiem na wszystkie metody), Dos Commands (dir, format, copy, device, etc.), Disk Copy (do szybkiego kopiowania dysków), Boot Selector (do zmieniania stacji bootującej dysk), poprawiony monitor z 80 znakowym ekranem i scrollowaniem w dwie strony!!! I, oczywiście, wszystkie stare opcje. Czy Datel'owi uda się kiedyś dorobić fontannę?!

- Katharsis powiększa liczbę swoich członków. Nowymi nabytkami są: Mr. Root (muzyk - ex. Pyrrusians), Kicia (muzyk - dawny użykownik Spectrum'a) i Jerry.

- Ami-Słownik sprzedawany przez firmę COMMPOL jest „zawiruszony” czyli ma wirusa! Jest nim Byte Bandit. Zanim więc wczytacie ten program zabijcie wiruska bo może poczynić szkody na dyskach i w waszym zdrowiu psychicznym.

Złote TILT'Y

Dzięki nieocenionej Cioci leży teraz na moim biurku grudniowy numer francuskiego miesięcznika TILT. Tak jak „C+VG” ma swój „HIT!”, ,,Amiga Jocker” swój „Ace Rated”; „CUAmiga” swój „Superstar”; ,,Power Play” swojego „Zero Hero”; tak „Tilt” przyznaje „TILT D'OR” co można przetłumaczyć jako Złote Tilt'y. Wręcza się je na corocznym przyjęciu, na które zapraszani są przedstawiciele wszystkich firm produkujących oprogramowanie. Oto jakie programy zostały nagrodzone w roku 1990.

- Animacja:

1. Prince Of Persia firmy Broderbund,
2. 4D Boxing firmy Mindscape,
3 Thunderstrike firmy Logotron.

- Grafika:

1. BAT firmy Ubi Soft,
2. Maupiti Island firmy Lankhor,
3. Unreal firmy Ubi Soft.

- Symulator lotu:

1. LHX Attack Chopper firrny Electronic Arts,
2. F-29 Retaliator firmy Ocean,
3. SU 25 Soviet Attack Fighter firmy Electronic Arts.

- Symulator (bez symulatorów lotu):

1. Wolfpack firmy Mirrorsoft,
2. Silent Service II firmy Microprose,
3. Tank firmy Electronic Arts.

- Gra przygodowa:

1. Maupiti Island firmy Lankhor,
2. Operation Stealth firmy Delphine,
3. Colonel's Request firmy Sierra.

- Gra fantasy:

1. Captive firmy Mindscape,
2. The Immortal firmy Electronic Arts,
3. Chaos Strikes Back firmy FTL.

- Program edukacyjny:

1. ADI firmy Cockteil Vision,
2. The Lift firmy Nathan,
3. Ecritures Automatiques firmy Jeriko.

- Gra strategiczno-zręcznościowa:

1. Ex-aequo Full Metal Planet firmy Infogrames i Powermonger firmy Electronic Arts,
3. Tower Of Babel firmy Microprose.

- Gra logiczno-zręcznościowa:

1. Pipemania firmy Empire/Titus,
2. Welltris firmy Mirrorsoft;
3. Klax firmy Tengen/Domark.

- Język programowania:

1. AMOS firmy Mandarin Software
2. Basic 1000D firmy Mori,
3. A-Debog firmy Arobace

- Program do tworzenia grafiki:

1. Arabesque firmy Upgrade;
2. Creator firmyApplications Systems,
3. Canvas firmy Microdeal.

- Program do tworzenia muzyki:

1. Ex-aequo Cubase firmy Steinberg/Sarro i Sequence 1000 firmy Fretless,
3. Big Boss firmy RNS.

- Symulacja sportowa:

1. Indianapolis 500 firmy Electronic Arts,
2. Kick Off II firmy Anco/Ubi;
3. Panza Kick Boxing firmy Loriciel.

- Wyscigi samochodowe:

1. Lotus Turbo Esprit Challenge firmy Gremlin
2. Super Monaco GP firmy Sega,
3. Ultimate Ride firmy Mindscape.

- Gra przygodowo-zręcznościowa:

1. Adventure of Link firmy Nintendo
2. Gold of The Aztecs firmy US Gold
3. Prince of Persia firmy Broderbund

- Gra zręcznościowa:

1. Ex-aequo Revenge of Shinobi firmy Sega i Turrican firmy Rainbow,
3. Fire and Brimstone firmy Firebird.

- Gra „Beat'em Al”:

1. PC Kid firmy Hudson Soft
2. Legendary Axe firmy Victor,
3. Ninja Warriors firmy Virgin.

- Gra „Shoot'em Up”:

1. Ex-aequo Battle Squadron firmy Innerprise/Ubi i Thunderforce III firmy Sega,
3. Simulcra firmy Microstyle,

- Najlepsze konwersje gier salonowych:

1. Pang firmy Ocean,
2. Ghouls'n Ghosts firmy Sega,
3. Golden Axe firmy Virgin.

- Nagroda Specjalna Jury:

Railroad Tycoonon firmy Microprose.

- Gra z przyszłością:

SW IV firmy Storm

Tak oceniało programy jury francuskiego miesięcznika „Tilt”. W numerze-dodatkowo-została zamieszczona wkładka ze zdjęciami i opisami nagrodzonych gier.

Pozostaje nam tylko czekać, aż te gry dotrą do Polski i mieć nadzieję, że Francuzi mają gust podobny do Polaków.

Mr. Raf

MAPA PAMIĘCI – CZĘŚĆ III

Przedstawiamy dalszy ciąg opisu szczegółowej mapy pamięci komputerów rodziny C16/116/Plus4. Poniżej prezentujemy obszar od komórki o adresie 161 do komórki o adresie 767 włącznie. Znajdują się w nim komórki robocze systemu operacyjnego i interpretera BASIC oraz stos mikroprocesora 7501.

Komórka pamięci o adresie 194 zawiera informację o sposobie wyświetlania kolejnego znaku. Jeżeli jej zawartość jest równa 18, to następne znaki aż do końca linii będą wyświetlane w inwersji. Normalne wyświetlanie przywraca wpisanie do tej komórki wartości 0. Wykonajmy przykładowy program:

```
10 POKE 194,18: PRINT "TEN TEKST";:
   PRINT "JEST W REWERSIE": PRINT "A TEN NIE"
```

Zawartość komórki o adresie 198 informuje o aktualnie wciśniętym klawiszu. Jeżeli jest równa 64, to nie jest naciśnięty żaden klawisz. Kody poszczególnych klawiszy można ustalić wykonując program:

```
10 PRINT PEEK(198): GOTO 10
```

Przy wciskaniu klawisza na ekranie jest wyświetlany jego kod. Przerwać program można klawiszem STOP. Jeżeli umieścimy w programie linie:

```
100 PRINT "NACISNIJ SPACJE"
110 IF PEEK(198) < > 60 TREN GOTO 110
```

to wykonywanie programu zostanie zatrzymane do momentu naciśnięcia spacji.

Komórki o adresach 202 i 205 zawierają współrzędne kursora tekstowego - odpowiednio kolumnę i wiersz w których zostanie wyświetlony następny znak. Wpisywanie do komórki o adresie 202 wartości z zakresu 0-39 przesuwa pozycję wydruku do odpowiedniej kolumny działając podobnie jak instrukcja TAB. Przykładowy program:

```
20 POKE 202,10: PRINT "SYMULACJA TAB"
```

wypisze tekst zawarty w instrukcji PRINT od kolumny 10. Zmiana wiersza wydruku nie jest taka prosta. Po wpisaniu nowej wartości do komórki o adresie 205 :

```
30 POKE 205,20: PRINT "PIONOWY TAB"
```

pierwszy znak wyprowadzonego później tekstu pojawia się na „starej" pozycji kursora, a dopiero następne w wierszu 20. Rozbicia tekstu można uniknąć wpisując jako jego pierwszy znak spację.

Zawartość komórki o adresie 240 decyduje o zatrzymywaniu przesuwania treści ekranu po wyprowadzeniu kolejnego wiersza wydruku. Można to wykorzysta, kiedy wykonywany program wyprowadza wiele wyników na ekran i istnieje niebezpieczeństwo usunięcia wydruku z ekranu zanim zostanie zanotowany przez operatora. Załóżmy, ze wykonywany program wyprowadza zawartość tablicy D(), która zawiera 40 danych:

```
 *
10 DIM D(40)
 *
100 SCNCLR: FOR J=1 TO 40
101 PRINT "WYNIK";J;D(J)
102 POKE 240,1
103 NEXT J
 *
```

Podczas wypisywania po każdej danej wydruk jest zatrzymywany i można go wznowić naciskając dowolny klawisz. Po usunięciu linii 102 wszystkie dane zostaną wyprowadzone w jednym ciągu i na ekranie pozostanie tylko 20 ostatnich. Zatrzymywanie wydruku po wypisaniu każdych 10 danych można osiągnąć umieszczając w powyższym programie linię 102 o treści:

```
102 IF J/10 = INT(J/10) THEN POKE 240,1
```

W komputerach Commodore 16/116/Plus4 konstruktorzy przewidzieli zainstalowanie kilku różnych pamięci stałych (ROM) jednocześnie. Fabrycznie jedynie w modelu Plus4 zastosowano dodatkowe pamięci ROM z oprogramowaniem użytkowym. Przestrzeli adresowa ROM (obszar 32768 - 65535 z wyłączeniem fragmentów, które zajmują rejestry urządzeń we/wy) została podzielona na dwie części - dolną i górną. W każdej z nich mogą zgłaszać sic cztery bloki pamięci ROM, co daje razem 16 możliwych kombinacji. Informacja o aktualnej konfiguracji pamięci ROM zawarta jest w komórce o adresie 251. Możliwe konfiguracje przedstawia tabela:

	BLOK DOLNY	BLOK GÓRNY
0	interpreter BASIC	system operacyjny (KERNAL)
1	wbudowane oprogramowanie (D)	system operacyjny (KERNAL)
2	moduł (cartridge) (D)	system operacyjny (KERNAL)
3	zarezerwowane (D)	system operacyjny (KERNAL)
4	interpreter BASIC	wbudowane oprogramowanie (G)
5	wbudowane oprogramowanie (D)	wbudowane oprogramowanie (G)
6	moduł (cartridge) (D)	wbudowane oprogramowanie (G)
7	zarezerwowane (D)	wbudowane oprogramowanie (G)
8	interpreter BASIC	moduł (cartridge) (G)
9	wbudowane oprogramowanie (D)	moduł (cartridge) (G)
10	moduł (cartridge) (D)	moduł (cartridge) (G)
11	zarezerwowane (D)	moduł (cartridge) (G)
12	interpreter BASIC	zarezerwowane (G)
13	wbudowane oprogramowanie (D)	zarezerwowane (G)
14	moduł (cartridge) (D)	zarezerwowane (G)
15	zarezerwowane (D)	zarezerwowane (G)

Interpreter BASIC pracuje jedynie w konfiguracji 0. Jeżeli wpiszemy do komórki o adresie 251 instrukcją POKE wartość różną od 0, to zablokujemy komputer. Zmieniać zawartość tej komórki pamięci można w ogra-

niczonym zakresie z poziomu monitora TEDMON. Po przejściu do monitora (instrukcja MONITOR) wykonujemy polecenie M FB wyświetlając zawartość fragmentu pamięci od adresu 251 ($FB szesnastkowo). Bezpiecznie można wpisywać pod ten adres tylko wartości 1 - 3, co powoduje odłączenie bloku pamięci ROM interpretera BASIC i przyłączanie innych bloków. W komputerach Plus4 po wpisaniu do komórki $FB liczby 1 pod adresami $8000- $BFFF można obejrzeć zawartość ROM'u wbudowanego oprogramowania. Po wpisaniu liczby 2 lub 3, jeżeli nie są podłaczone dodatkowe moduły, polecenie M 8000 wyświetla przypadkowe wartości odczytane z szyny danych. Przed powrotem do interpretera BASIC (instrukcja X monitora) należy koniecznie wpisać pod adres $FB wartość 0.

Adres HEX	Adres DEC	Etykieta	Opis
00A1-00A2	161-162	T2	pamięć chwilowa
00A3-00A5	163-165	TIME	zegar czasu rzeczywistego (ok. 1/60s)
00A6	166	R2D2	rejestr łącza szeregowego
00A7	167	TPBYTE	rejestr procedur obsługujących magnetofon
00A8	168	BSOUR1	rejestr łącza szeregowego
00A9	169	FP VERR	chwilowy wektor barwy
00AA	170	DCOUNT	rejestr procedur obsługujących magnetofon
00AB	171	FNLEN	długość aktualnej nazwy zbioru
00AC	172	LA	logiczny numer zbioru
00AD	173	SA	aktualny adres wtórny
00AE	174	FA	aktualny numer urządzenia
00AF-00B0	175-176	FNADR	wskaźnik: aktualna nazwa zbioru
00B1	177	ERRSUM	licznik błędów odczytu z taśmy
00B2-00B3	178-179	STAL	adres startu we/wy
00B4-00B5	180-181	MEMUSS	bazowy adres ładowania
00B6-00B7	182-183	TAPEBS	adres końca ładowania (kaseta)
00B8-00B9	184-185	TMP2	adres dla VECTOR
00BA-00BB	186-187	WRBASE	wskaźnik znaku w buforze magnetofonu
00BC-00BD	188-189	IMPARM	
00BE-00BF	190-191	FETPTR	rejestr procedury LONG-FETCH
00C0-00C1	192-193	SEDSAL	rejestr dla przesuwania zawartości ekranu (scroll)
00C2	194	RVS	wskaźnik: znak w trybie odwróconym (revers); $12=tak, $00=nie
00C3	195	INDX	wskaźnik: koniec wiersza logicznego (INPUT)
00C4-00C5	196-197	LSXP	pozycja X/Y kursora (INPUT)
00C6	198	SFDX	wskaźnik: naciśnięty klawisz ($40=żaden)
00C7	199	CRSW	wskaźnik: INPUT lub GET z klawiatury
00C8-00C9	200-201	PNT	wskaźnik: aktualny wiersz ekranu

Adres HEX	Adres DEC	Etykieta	Opis
00CA	202	PNTR	kolumna kursora w aktualnym wierszu
00CB	203	QTSW	wskaźnik: edytor w trybie cudzysłowu (0=nie)
00CC	204	SEDT1	długość aktualnego wiersza ekranu
00CD	205	TBLX	wiersz w którym znajduje się kursor
00CE	206	DATAX	ostatni znak (we/wy)
00CF	207	INSRT	wskaźnik: tryb INST, 0=ilość wstawek
00D0-00D7	208-215		zarezerwowane dla syntetyzatora mowy
00D8-00E8	216 -232		zarezerwowane dla oprogramowania użytkowego
00E9	233	CIRSEG	obszar roboczy(CIRCLE)
00EA-00EB	214-235	USER	kolor i jasność kursora
00EC-00EE	236-238	KEYTAB	wektor. tabela dekodowania klawiatury $E026
00EF	239	NDX	ilość znaków w buforze klawiatury
00F0	240	STPFLG	wskaźnik: zatrzymanie przesuwania zawartości ekranu i programu (CTRL-S)
00Fl-00F2	241-242	TO	rejestr dla monitora maszynowego
00F3	243	CHRPTR	adres zero-page dla monitora
00F4	244	BUFEND	adres zero-page dla monitora
00F5	245	CHKSUM	rejestr dla sumy kontrolnej
00F6	246	LENGTH	
00F7	247	PASS	
00F8	248	TYPE	typ bloku
00F9	249	USEKDY	(b7 = 1) dla write, (b6 = 1) dla read
00FA	250	XSTOP	rejestr dla X przy teście klawisza STOP
00FB	251	CURBNK	aktualna konfiguracja banków ROM
00FC	252	XON	znak do wysłania dla X-On
00FD	253	XOFF	znak do wysłania dla X-Off
00FE	254	SEDT2	pamięć robocza edytora
00FF	255	LOFBUF	
0100-010F	256-271	FBUFFR	
0110	272	SAVEA	pamięć pośrednia dla SAVE i RESTORE
0111	273	SAVEY	pamięć pośrednia dla SAVE i RESTORE
0112	274	SAVEX	pamięć pośrednia dla SAVE i RESTORE
0113-0122	275-289	COLKEY	tabela barwa/jasność w RAM
0123	290		
0124-01FF	291-511	SYSSTK	stos procesora
0200-0258	512-600	BUF	systemowy bufor wejściowy
0259-025A	601-602	OLDLIN	poprzedni numer wiersza BASIC
025B-025C	603-604	OLDTXT	wskaźnik: instrukcja BASIC dla CONT
025D-02AC	605-684		obszar roboczy BASIC/DOS

Adres HEX	Adres DEC	Etykieta	Opis
025D	605	XCNT	licznik pętli DOS
025E-026D	606-621	FNBUFR	obszar dla nazw zbiorów
026E	622	DOSFIL	długość 1 nazwy zbioru
026F	623	DOSDSl	DOS (napęd 1)
0270-0271	624-625	DOSF1A	adres 1 nazwy zbioru
0272	626	DOSF2L	długość 2 nazwy zbioru
0273	627	DOSFA	DOS (napęd 2)
0274-0275	628-629	DOSF2A	adres 2 nazwy zbioru
0276	630	DOSLA	DOS adres logiczny
0277	631	DOSFA	DOS adres urządzenia
0278	632	DOSSA	DOS adres wtórny
0279-027A	633-634	DODDID	DOS ID dyskietki
027B	635	DIDCHK	wskaźnik ID
027C	636	DOSSTR	bufor wyjściowy DOS
027D-02AC	637-684	DOSSPC	obszar roboczy DOS
02AD-02AE	685-686	XPOS	aktualna pozycja X kursora grafiki
02AF-02B0	687-688	YPOS	aktualna pozycja Y kursora grafiki
0281-02B2	689-690	XDEST	współrzędna docelowa X
02B3-02134	691-692	YDEST	współrzędna docelowa Y
02B5-02CB	693-715		obszar roboczy procedur graficznych
02CC-02E3	716-739		obszar roboczy procedur graficznych oraz PRINT USING
02E4	740	CHRPAG	MSB adresu generatora znaków dla rozkazu CHAR
02E5	741	BITCNT	rejestr dla GSHAPE
02E6	742	SCALEM	wskaźnik: tryb SCALE ($00=wył)
02E7	743	WIDTH	wskaźnik: podwójna wielkość pixela
02E8	744	FILFLG	wskaźnik: rysowanie prostokąta (rozkaz BOX)
02E9	745	BITMSK	pamięć chwilowa maski bitów
02EA	746	NUMCNT	długość łańcucha
02EB	747	TRCFLG	wskaźnik: tryb TRACE ($00=wył)
02EC	748	T3	pamięć pośrednia dla DIRECTORY
02ED-02EE	749-750	T4	pamięć pośrednia dla DIRECTORY
02EF	751	VTEMP3	pamięć chwilowa dla grafiki
02F0	752	VTEMP4	ilość parametrów grafiki
02F1	753	VTEMP5	parametr: $01=względny, $00=bezwzgl.
02F2-02F3	754-755	ADRAY1	wskaźnik dla procedury zamiany liczby zmiennoprzecinkowej w całkowitą
02F4-02F5	756-757	ADRAY2	wskaźnik dla procedury zamiany liczby całkowitej w zmiennoprzecinkową
02F6-02FD	758-765		nie używane
02FE-02FF	766-767	BNKVEC	wektor dla modułu (cartridge)

Wojciech Wasilewski

Najprostszy „Old”

W pierwszym numerze naszego pisma wydrukowana została procedura realizująca operację OLD, czyli odzyskanie pomyłkowo skasowanego (instrukcją NEW lub przez użycie klawisza RESET) programu w BASIC'u. Ma ona jednak poważną wadę, jak zresztą wszystkie procedury OLD ładowane z dysku czy z taśmy. Otóż, aby za pomocą tej procedury odzyskać nasz omyłkowo skasowany program, musimy być już wcześniej przygotowani na taką ewentualność i przed rozpoczęciem pracy załadować procedurę do pamięci komputera, tak aby była ona już tam obecna w chwili, gdy okaże się potrzebna.

Jak uczy doświadczenie, przypadki wymagające użycia operacji OLD zdarzają się akurat wtedy, gdy absolutnie się tego nie spodziewamy. W takiej sytuacji przydałaby się procedura OLD w pamięci ROM komputera, tak aby móc do niej sięgnąć w każdej chwili, bez potrzeby ładowania dodatkowych programów. I procedura taka istotnie się tam znajduje. Do odzyskiwania skasowanych programów można wykorzystać pewne wewnętrzne procedury interpretera BASIC'a. Po skasowaniu programu należy wpisać:

```
POKE 4097,1 : SYS 34840 : SYS 34891
```

i nasz stracony program jest z powrotem do dyspozycji. Oczywiście, jeżeli nasz program znajdował się w pamięci pod innym adresem niż standardowy ($1001 = dziesiętnie 4097), musimy wcześniej wpisać odpowiedni adres do komórek 43 (młodszy bajt) i 44 (starszy bajt), będą-cych wskaźnikiem początku programu w BASIC'u, i tego też adresu użyć w instrukcji POKE zamiast 4097. Np. programy korzystające; z grafiki najczęściej lokują się w pamięci pod adresem $4001 (dziesiętnie 16385), tak więc dla odtworzenia programu musimy wykonać:

```
POKE 43,1 : POKE 44,64
POKE 16385,1 : SYS 34840 : SYS 34891
```

J.R.

Automatyczne wygaszanie ekranu

Program nagrodzony główną nagrodą w naszym konkursie - Najlepszy Program Miesiąca.

Automatyczne wygaszanie ekranu, gdy przez określony czas nie pracujemy na komputerze (nie używamy jego klawiatury), jest niezwykle popularną funkcją w oprogramowaniu komputerów klasy PC. Nic nie stoi na przeszkodzie, aby funkcję tą zastosować i w naszym Commodore. W tym przypadku wygaszanie ekranu ma dodatkową zaletę poza oszczędzaniem monitora: ponieważ szybkość przetwarzania Commodore przy wyłączonym ekranie jest większa, przyspiesza to np. programy wykonujące długotrwałe obliczenia.

Przedstawiony na listingu program po uruchomieniu instaluje się rezydentnie w pamięci i automatycznie wygasza ekran po 1 min. braku aktywności klawiatury. Czas ten można zmienić podając odpowiedni parametr w wywołaniu programu - najpierw program należy wyłączyć instrukcją SYS 1660, a następnie ponownie włączyć wydając instrukcję SYS 1622,n co spowoduje wygaszanie ekranu po czasie ok. n*2,6 s. Ponowne włączenie wygaszonego ekranu następuje w momencie wciśnięcia jakiegokolwiek klawisza; można je uzyskać również programowo wpisując do komórki 198 dowolną wartość z zakresu 0-63 (użyteczne np. w programach wykonujących długie obliczenia, które mogą automatycznie włączać ekran po ich zakończeniu).

Program znajduje się w obszarze pamięci $600-688 (dziesiętnie 1536-1672) i wykorzystuje jako obszar roboczy komórki $E0-E4 (224-228), tak więc nie należy równocześnie z nim używać innych programów wykorzystujących ten sam obszar pamięci. Program modyfikuje systemowy wektor przerwań IRQ znajdujący się w komórkach $314-315, co może uniemożliwić równoczesne działanie ewentualnych innych programów rezydentnych wykorzystujących ten wektor.

Jarosław Rafa

```
1 DATA A5,E0,D0,34,A5,C6,C9,40
2 DATA F0,09,A9,00,85,E1,85,E2
3 DATA 4C,0E,CE,E6,E1,D0,F9,E6
4 DATA E2,A5,E2,C5,E4,D0,F1,AD
5 DATA 19,FF,85,E3,A9,00,8D,19
6 DATA FF,AD,06,FF,29,EF,8D,06
7 DATA FF,A9,01,85,E0,4C,0E,CE
8 DATA A5,C6,C9,40,F0,F7,AD,06
9 DATA FF,09,10,8D,06,FF,A5,E3
10 DATA 8D,19,FF,A9,00,85,E0,85
11 DATA E1,85,E2,4C,0E,CE,A0,00
12 DATA 20,A5,04,C9,2C,D0,08,20
13 DATA 81,9D,86,E4,4C,6B,06,A9
14 DATA 17,85,E4,A9,00,85,E0,78
15 DATA A2,00,A0,06,8E,14,03,8C
16 DATA 15,03,58,60,78,A2,0E,A0
17 DATA CE,8E,14,03,8C,15,03,58,60
20 PRINT"<CLEAR> <DOWN> <DOWN>
   SUPERKEY (C) TOOLSOFT'87"
30 PRINT "<DOWN> <DOWN> TEN PROGRAM
   WYGASZA EKRAN, JEZELI PRZEZ": PRINT
   PEWIEN CZAS NIE BYL NACISKANY ZADEN"
40 PRINT "KLAWISZ, PO NACISNIECIU
   JAKIEGOKOLWIEK": PRINT "KLAWISZA EKRAN
   PONOWNIE SIE WLACZA."
50 PRINT"<DOWN> <DOWN> WYWOLANIA:":
   PRINT "<DOWN> <RIGHT> <RIGHT><RIGHT>
   <RVSON>SYS 1622<RVSOFF> - WYGASZENIE
   PO CZASIE 1 MIN"
60 PRINT "<DOWN> <RIGHT> <RIGHT>
   <RIGHT> <RVSON>SYS 1622,N<RVSOFF>
   - WYGASZENIE PO CZASIE":PRINT TAB(16)
   "<RVSON>N*2.6 S'
70 PRINT "<DOWN> <RIGHT> <RIGHT>
   <RIGHT> <RVSON>SYS 1660<RVSOFF>
   - WYLACZENIE PROGRAMU
100 FOR I=1536 TO 1672
110 READ P$:POKE I,DEC(P$):NEXT
120 PRINT "<DOWN> <DOWN> <RVSON>
   PROGRAM AKTYWNY ! ":SYS 1622
```

Listing programu

Uczymy się grać

Wiemy już, że Commodore 116, 16 lub plus4 może z powodzeniem realizować nasze muzyczne pasje, warto więc poświęcić trochę uwagi formie notacji dźwięków. Wiedza ta ułatwi nam przenoszenie motyki z partytury do programu komputerowego.

Przypominam nazwy oktaw klawiatury fortepianu:

1. Subkontra (trzy dźwięki).
2. Kontra.
3. Wielka.
4. Mała.
5. Razkreślna.
6. Dwukreślna.
7. Trzykreślna.
8. Czterokreślna.
9. Pięciokreślna (jeden dźwięk).

W zapisie nutowym dźwięki oktaw pierwszej (subkontry) do częściowo piątej zapisywane są na pięciolinii z kluczem basowym, pozostałe z kluczem wiolinowym. Jak pamiętamy, rodzina C-16 emituje dźwięki od ostatnich trzech tonów oktawy wielkiej do pierwszego tonu oktawy pięciokreślnej. W zapisie nutowym wygląda to następująco:

Znak nazywany krzyżykiem (♯) podwyższa o pół tonu wartość zapisanego za nim dźwięku. Inny znak - bemol (♭) obniża o pół tonu zapisany za nim dźwięk.

Linia przerywana nad pięciolinią (lub pod nią) oznacza, ze zapisane dźwięki są wyższe (lub niższe) i występują na następnej, wyżej (lub niżej) narysowanej pięciolinii. W ten prosty sposób unika się graficznego rozbudowywania pięciolinii.

Chcąc przenosić zapis nutowy do programu komputerowego, musimy poznać oznaczenia graficzne nut w odniesieniu do czasu trwania dźwięku. Oznaczenia te pozostają w ścisłej zależności od nuty podstawowej, nazwanej nutą całą. Czas trwania nuty całej wyznacza urządzenie zwane metronomem lub taktomierzem. Urządzenie to sygnalizuje w ciągu jednej minuty 112,5 taktów. Nasz komputer realizuje w ciągu minuty 3600 taktów mikroprocesora. Łatwo możemy obliczyć, że na jeden takt metronomu przypadają 32 takty mikroprocesora (3600:112,5=32). Inaczej mówiąc - pełny takt muzyczny w naszym komputerze trwa 64 jednostki czasowe (takty mikroprocesora), co równa się czasowi brzmienia całej nuty.

Odbiór muzyki jest odczuciem subiektywnym, co pozwala zmieniać ten podstawowy parametr, tj. zwięk-

7 64 118 169 217 262 305 345 383 419 453 485 516 544 571 596 620 643 664 685 704 722 739

A ais H C cis d dis e f fis g gis a ais h c1 cis d1 dis e1 f1 fis g1

755 770 784 798 810 822 834 844 854 864 873 881 889 897 904 911 917 923 929 934 939 944 949

gis a1 ais h1 c2 cis d2 dis e2 f2 fis g2 gis a2 ais h2 c3 cis d3 dis e3 f3 fis

953 957 960 964 966 970 973 976 978 981 984 986 988 990 992 993 995 997

g3 gis a3 ais h3 c4 cis d4 dis e4 f4 fis g4 gis a4 ais h4 c5

szać lub zmniejszać czas brzmienia nuty całej według własnych upodobań.

Oznaczenia nut i czasy ich trwania ilustruje tabela.

L.p.	Nazwa nuty	Znak graficzny	Czas trwania nuty – ilość taktów mikroprocesora
1	**Nuta cała**		**60**
2	**Półnuta**		**30**
3	**Ćwierćnuta**		**15**
4	**Ósemka**		**7.5**
5	**Szesnastka**		**3.25**
6	**Trzydziestkadwójka**		**1.625**

Spotykana w zapisie nutowym kropka z prawej strony nuty wydłuża czas jej brzmienia o połowę.

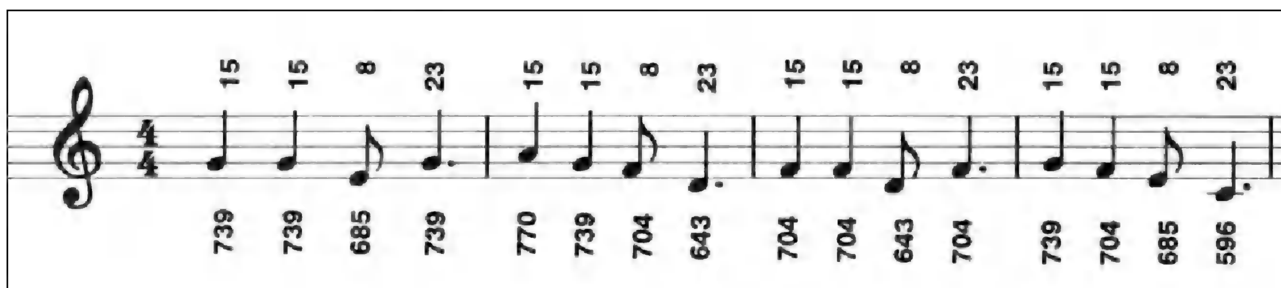
Dla utrwalenia naszych wiadomości spróbujemy przenieść do programu komputerowego pierwsze takty piosenki „Lato” (żeby było nam cieplej) napisanej przez Witolda Krzemieńskiego dla potrzeb filmu „Szatan z VII klasy”.

brzmienia nuty, choć jak powiedziano wyżej, według własnych odczuć możemy i ten parametr zmieniać.

Problem pauzy w programie rozwiążemy dźwiękiem o wartości 1022. Dźwięk ten leży na pograniczu możliwości głosowych komputera i praktycznie nie jest już słyszalny - uznajemy więc go za „ciszę”.

Prezentowany listing jest finałem naszej dotychczasowej pracy. Program uruchamiamy rozkazem RUN.

```
  0 REM PIERWSZE TAKTY PIOSENKI "LATO"
 10 VOL 8
 20 DO
 30 READ X,Y
 40 SOUND 1,X,Y
 50 REM WYZNACZENIE KONCA WYBIERANIA
 60 REM DZWIEKOW I PAUZ
 90 LOOP UNTIL X=596 AND Y=24
100 END
110 REM WARTOSCI I CZASY TRWANIA DZWIEKU
120 REM I PAUZ
150 DATA 739,16,1022,16,739,16,1022,16
160 DATA 685,8,1022,8,739,24,1022,24
170 DATA 770,16,1022,16,739,16,1022,16
180 DATA 704,8,1022,8,643,24,1022,24
190 DATA 704,16,1022,16,704,16,1022,16
200 DATA 643,8,1022,8,704,23,1022,24
210 DATA 739,16,1022,16,704,16,1022,16
220 DATA 685,8,1022,8,596,24
```



Na zapis nutowy tej melodii nanosimy wartości nut, oraz ustalone czasy brzmienia poszczególnych dźwięków.

Istotnym elementem muzycznym są pauzy pomiędzy dźwiękami. Czas ich trwania z zasady odpowiada czasowi

Życzymy miłej i pożytecznej zabawy.

Jan Siedlecki

„64 plus 4" ogłasza szybko i tanio !

Przedsiębiorstwo ABUK S-ka z o.o. oferuje państwu szybką i tanią obsługę reklamową. Ogłoszenia drobne od osób indywidualnych (do 10 słów) przyjmujemy bezpłatnie. Większe - 1000 zł za słowo.

Reklamy ramkowe (minimalny format - 20 cm^2):

- **1 cm^2 ogłoszenia - 4500 zł**
- **cała strona - 2,5 mln zł**
- **kolor - odpowiednio 100% drożej.**

Ogłoszenia przyjmujemy za pośrednictwem poczty.
Nasz adres :

„64 plus 4"
85-166 Bydgoszcz 43
skrytka pocztowa 64

Treść ogłoszenia z określeniem formatu reklamy (ewentualnie zamówieniem koloru) prosimy nadsyłać listem poleconym wraz z odcinkiem wpłaty (za pomocą przekazu pieniężnego) na konto Przedsiębiorstwa ABUK Bank Polska Kasa Opieki SA Oddział w Bydgoszczy, konto nr : 5.09011-400522.7-136-11-111.0
Dołączenie do zamówienia odcinka wpłaty przyspieszy zamieszczenie reklamy o miesiąc.

Przeliczamy adresy

Krótko o po kupieniu sobie C-64 odczułem potrzebę posiadania programu, dzięki któremu mógłbym przeglądać pełne 64 kB pamięci. Były to czasy, w których cartridge takie jak Final II lub Action Replay były tylko niespełnionym marzeniem. Wszystkie monitory, które posiadałem, przy próbie odłączenia ROM-ów od $D000 w górę wieszały system. Postanowiłem więc napisać program, który rozwiązywał ten problem.

Załóżmy, że adres który chcemy wyświetlić na ekranie znajduje się w komórkach $FB i $FC (w FB bajt młodszy, a w FC starszy). Jeśli tym adresem jest np. $FD30 to w $FB znajduje się wartość $B0 a w $FC wartość $FD. Jak wyświetlić to na ekranie ? Przede wszystkim należy te wartości (każdą połówkę bajtu) zamienić na kody ekranowe, które potem można spokojnie przepisać na ekran. W naszym przykładzie wartości ekranowe będą wyglądały następująco:

$	F	D	3	0
	$06	$04	$33	$30

Pierwsza myśl, która mi się nasunęła, to zrobić krótką, bo mającą 16 bajtów, tablicę wartości, a potem tylko wybierać je i umieszczać na ekranie. Napisałem więc procedurę :

```
        LDA     $FC         ;pobranie bajtu
        ISR     L1          ;przeliczenie na kod
                            ;ekranowy
        STA     $079C       ;zapamiętanie na ekranie
        LDA     $FC         ;ponowne pobranie bajtu
        AND     #$0F        ;uzyskanie tylko młodszej
                            ;części bajtu
        ISR     :L2         ;przeliczenia
        STA     $079D       ;zapamiętanie na ekranie
        LDA     $FB         ;to samo
        ISR     :L1         ;dla młodszej połówki
        STA     $079E       ;adresu
        LDA     $FB
        AND     #$0F
        ISR     :L2
        STA     $079F
        RTS                 ;powrót
:L1     LSR                 ;uzyskanie starszej
        LSR                 ;połówki bajtu
        LSR
        LSR
        LSR
        LSR
:L2     TAX
        LDA     TAB,X       ;pobranie wartości z tablicy
        RTS
:TAB    DC,B    $30,$31,$32,$33,$34,$35,$36,$37
                $38,$39,$01,$02,$03,$04,$05,$06
```

Komórki od $079C do $079F przedstawiają pamięć ekranu i można je dowolnie zmieniać (muszą jedynie tworzyć ciąg). Taki sposób przeliczania jest szybki ale mało elegancki i długi. Postanowiłem więc napisać procedurę, która „autentycznie” liczyłaby adresy. Początek jest taki sam. Zmiany następują dopiero pod adresem :L2.

```
:L2     CLC
        CMP     #$0A        ;czy wartość jest równa
                            ;czy większa od $0A
        BCS     :L3         ;tak
        ADC     #$30        ;nie
        RTS
:L3     SBC     #$09
        RTS
```

Odpadła konieczność stosowania tablicy wartości i w wyniku tego procedura skróciła się o kilka bajtów. W ten sposób rozwiązałem swój pierwszy problem i już spokojnie mogłem napisać program który nosi nazwę „SCANNER” i służy do przeglądania pełnych 64 kB pamięci.

Zmiany adresów dokonuje się przez :

F1 + $300
F3 – $01
F5 + $01
F7 – $100

A oto sam program znajdujący się pod adresem $1000 (4096):

```
$1000   JSR     $E544
        LDX     #$00
        STX     $FB
        STX     $FC
        LDA     #$80
        STA     $028A
        LDX     #$26
$1010   LDA     $10B0,X
        STA     $0798,X
        DEX
        BPL     $1010
$1019   JSR     $1056
        JSR     $107C
$101F   JSR     $FFE4
        CMP     #$88
        BNE     $1028
        DEC     $FC
        JMP     $1019
$102B   CMP     #$85
        BNE     $1038
        INC     $FC
        INC     $FC
        INC     $FC
        JMP     $1019
$1038   CMP     #$87
        BNE     $1045
        INC     $FB
        BNE     $1042
        INC     $FC
$1042   JMP     $1019
$1045   CMP     #$86
        BNE     $101F
        DEC     $FB
        LDA     $FB
        CMP     #$FF
        BNE     $1042
        DEC     $FC
        JMP     $1019
$1056   SEI     RTS
        INC     $01
        LDA     $FC
        PHA
        LDX     #$03
$105E   LDY     #$00
$1000   LDA     ($FB),Y
        STA     $0400,Y
        INY
        BNE     $1060
        INC     $FC
        INC     $1064
        DEX
        BNE     $105E
        LDA     #$04
        STA     $1064
        PLA
        STA     $FC
        DEC     $01
        CLI
        RTS
$107C   LDA     $FC
        JSR     $10A1
        STA     $079C
        LDA     $FC
        AND     #$0F
        JSR     $10A5
        STA     $079D
        LDA     $FB
        JSR     $10A1
        STA     $079E
        LDA     $FB
        AND     #$0F
        JSR     $10A5
        STA     $079F
        RTS
$10A1   LSR
        LSR
        LSR
        LSR
$10A5   CLC
        CMP     #$0A
        BCS     $10AD
        ADC     #$30
        RTS
$10AD   SBC     #$09
        RTS

$10B0:
0D,05,0D,24,20,20,20,20
20,13,13,01,0E,0E,05,12
20,02,19,20,08,09,2D,0D
01,0E,20,27,03,36,34,2D
10,0C,15,13,20,34,27
```

Sambor Kuźma

Graj aby wygrać

„THE GREAT GIANA SISTERS" - Znane są Wam zapewnie przygody miłych sióstr Giany i Marii w labiryntach pełnych przedziwnych stworów. Jeżeli chcesz zabawę uczynić bardziej atrakcyjną, posłuż się POKE'ami :

POKE 53277,255	- sprite'y będą większe,
POKE 53277,80	- co drugi spńte będzie większy,
POKE 7450,96	- więcej czasu,
POKE 2447,100	- więcej „żyć",
POKE 6664,96	- mosty nie zapadają się,
POKE 7326,173	- nie kończąca się ilość diamentów przy rozbijaniu muru,
POKE 2213,164	- Giana jest punkiem i może strzelać,
POKE 3732,28	- zmiana koloru chmur,
POKE 5112,0	- Giana spada z góry na pole gry.

Może znasz jeszcze jakieś inne? Napisz !

Po wprowadzeniu poprawek grę wystartuj rozkazem SYS 2112.

„DEFLECTOR" - Słyszałeś zapewne o potędze lasera. Jeżeli chciałbyś stać się operatorem tego urządzenia i znasz pewne prawo fizyczne z działu optyki o odbiciu światła polecam ci grę „DEFLECTOR".

Pierwsze poziomy gry zapewne będą dla ciebie łatwe,ale dalsze...

Jeżeli chcesz potrenować i opanować laser tak, aby na egzaminie wypaść bezbłędnie, podsunę ci trochę ściągaczek: po załadowaniu gry do pamięci i zresetowaniu komputera wpisz poszczególne interesujące cię POKE'i i wystartuj grę rozkazem SYS 9658.

POKE 1023,1	- możesz wybrać odpowiedni poziom klawiszami „+" i „–",
POKE 1023,4	- nie pojawi się GREMLINS,
POKE 13967,165	- nie tracisz energii,
POKE 14073,165	- nie spalisz lasera,
POKE 11890,x	- wybierasz sobie poziom gry np. gdy chcesz poziom 13 za x podstawiasz 12 itd.

P O W O D Z E N I A !

Mr. Fox

Zapraszamy wszystkich do udziału w stałym konkursie pod hasłem:

Najlepszy program miesiąca

W konkursie udział mogą brać wszyscy, którzy nadeślą własne, nigdzie nie publikowane prace.
Tematyka programów dowolna.
Konkurs rozgrywany jest osobno dla komputerów C-16 i C-64.
Teksty programów należy nadsyłać na adres redakcji na taśmie magnetofonowej, dyskietce lub w postaci czytelnego rękopisu (dyskietki i taśmy będą przez redakcję zwracane).
Objętość programu wraz z opisem i komentarzem nie powinna przekraczać 4 stron maszynopisu.
Raz w miesiącu Sąd Konkursowy wybierze najlepsze programy przyznając ich autorom dwie główne nagrody po **500.000** zł każda. Decyzje Sądu Konkursowego są nieodwołalne. Oprócz zdobycia głównej nagrody autorzy mają szansę na publikację swych prac na łamach naszego pisma.
Pracę prosimy podpisywać imieniem i nazwiskiem oraz dokładnym adresem autora.

Redakcja

Scroll

Każdy z Was, Drodzy Czytelnicy na pewno widział nie jeden raz taki właśnie sposób wygodnej prezentacji tekstu na ekranie swojego C-64. Nie każdy jednak wie, jak coś takiego zrobić. Prezentowana obok procedura jest chyba najprostszym sposobem realizacji scrollingu. Aby nie tracić cennego czasu przystąpmy do opisu programu.

$1000-1027 - włączenie przerwań graficznych, zmiana wektora przerwań, ustawienie początku tekstu.

$1029-1043 - obsługa przerwań, w tym modyfikacja komórki $D016, która jest odpowiedzialna za sprzętowy scrolling poziomy oraz wywołanie procedury scroll'a. Przerwanie kończy się potwierdzeniem następnego przerwania graficznego i skokiem do procedury przerwań w pamięci ROM.

$1044-1073 - procedura przesuwająca tekst: zmniejszenie wartości dla scroll'a sprzętowego ($D016) i sprawdzenie, czy nastąpiło już maksymalne sprzętowe „wysunięcie". Jeżeli tak to przesunięcie całej linii znakowej o jedną literę w lewo i dostawienie nowego znaku z tekstu.

$1075-107D - procedura ustawiająca początek tekstu (na $1080).

$1080-... - miejsce na tekst zapisany w standardzie ASCII. Wartość zero powoduje ponowne ustawienie wskaźnika tekstu na jego początek („zapętlenie").

Przy wpisywaniu tego programu warto użyć assemblera lub dowolnego monitora z opcją assemblacji. Bardzo ważne jest dokładne prześledzenie działania programu zarówno od strony sprzętowej jak i zastosowanych technik adresowania.

Powodzenia w doświadczeniach !

Paweł Sołtysiński

```
.,1000  78        SEI
.,1001  A9 7F     LDA  #$7F
.,1003  8D 0D DC  STA  $DC0D
.,1006  A2 00     LDX  #$00
.,1008  8E 0E DC  STX  $DC0E
.,100B  E8        INX
.,100C  8E 1A D0  STX  $D01A
.,100F  A9 1 B    LDA  #$1 B
.,1011  8D 11 D0  STA  $D011
.,1014  A9 31     LDA  #$31
.,1016  8D 12 D0  STA  $D012
.,1019  A9 29     LDA  #$29
.,101B  8D 14 03  STA  $0314
.,101E  A9 10     LDA  #$10
.,1020  8D 15 03  STA  $0315
.,1023  20 75 10  JSR  $1075
.,1026  58        CLI
.,1027  60        RTS
.,1028  00        BRK
.,1029  A9 05     LDA  #$05
.,102B  8D 16 D0  STA  $D016
.,102E  A9 3B     LDA  #$3B
.,1030  CD 12 D0  CMP  $D012
.,1033  D0 FB     BNE  $1030
.,1035  A9 C8     LDA  #$C8
.,1037  8D 16 D0  STA  $D016
.,103A  2044 10   JSR  $1044
.,103D  EE 19 D0  INC  $D019
.,1040  4C 31 EA  JMP  $EA31
.,1043  00        BRK
.,1044  CE 2A 10  DEC  $1
.,1047  30 01     BMI  $104A
.,1049  60        RTS
.,104A  A9 07     LDA  #$07
.,104C  8D 2A 10  STA  $102A
.,104F  A2 00     LDX  #$00
.,1051  BD 01 04  LDA  $0401,X
.,1054  9D 00 04  STA  $0400,X
.,1057  E8        INX
.,1058  E027      CPX  #$27
.,105A  D0 F5     BNE  $1051
.,105C  A0 00     LDY  #$00
.,105E  81 FB     LDA  ($FB),Y
.,1060  D0 06     BNE  $1068
.;1062  20 75 10  JSR  $1075
.,1065  4C 5E 10  JMP  $105E
.,1068  29 3F     AND  #$3F
.,106A  8D 27 04  STA  $0427
.,106D  E6 FB     INC  $FB
.,106F  D0 02     BNE  $1073
.,1071  E6 FC     INC  $FC
.,1073  60        RTS
.,1074  00        BRK
.,1075  A9 80     LDA  #$80
.,1077  85 FB     STA  $FB
.,1079  A910      LDA  #$10
.,107B  85 FC     STA  $FC
.,107D  60        RTS
.:1080  43 36 34 2F 2B 34 20 26
.:1088  20 41 4D 49 47 41 21 20
.:1090  20 20 20 20 00 00 00 00
.:1098  00 00 00 00 00 00 00 00
```

Listing programu SCROLL

Ogonki i C-64

Wielu posiadaczy C-64 zapewne z zazdrością spogląda na inne komputery, w których mechanizm instalacji własnego kształtu liter jest o wiele prostszy niż w posiadanym przez nich komputerze. Przykładów nie trzeba daleko szukać, dość by wspomnieć o poczciwym Spektrum czy Amstradzie, gdzie operacji takiej dokonuje się przy użyciu specjalnych funkcji interpretera BASIC. W Commodore 64 zmiana kroju pisma jest również możliwa, ale wymaga od użytkownika pewnego zasobu wiedzy i praktycznych umiejętności. Warto jednak je zdobyć, wyniki będą na pewno satysfakcjonujące.

Na początek informacje bez których dalsze wyjaśnienia będą niezrozumiałe:

– generowaniem obrazu na ekranie monitora (lub OTV) zajmuje się w C-64 specjalizowany układ graficzny VIC (video interface chip);
– do opisania pełnego zestawu znaków (256) potrzeba 2048 bajtów czyli $0800 szesnastkowo;
– po włączeniu komputera VIC interpretuje dane dla znaków, które znajdują się w pamięci ROM od adresu $000 do $D7FF, a po wciśnięciu kombinacji C= + SHIFT (małe litery - tryb biurowy) od adresu $D800 do $DFFF;
– położenie aktualnego ekranu tekstowego jak i generatora znaków (tzn. danych o wyglądzie naszych liter) jest zmienne, co oznacza, że można je umieścić w wybranym przez nas miejscu. Należy jednak pamiętać, że zarówno ekran, jak i generator znaków muszą znajdować się w jednym z czterech szesnastokilobajtowych banków, na które (dla potrzeb VIC'a) została podzielona pamięć w C-64;
– całość generatora znaków można podzielić na cztery zbiory : 64 znaki standartowe, 64 „z shiftem”, 64 „inverse video” i 64 „inverse video” + „shift”, przy czym każdy z tych zbiorów zajmuje $0200 (512 bajtów dziesiętnie);
– zdefiniowane znaki (przez siebie lub pobrane z ROM) mogą być wyświetlone na ekranie jedynie w trybie tekstowym, czyli tym, który jest standardowo załączany po uruchomieniu komputera.

A teraz więcej wyjaśnień. Tryb tekstowy to taki tryb pracy układu wizyjnego, w którym generowany na ekranie obraz jest odwzorowaniem pewnego obszaru z pamięci, gdzie każdej widocznej na monitorze literze odpowiada dokładnie jeden bajt. Podczas pracy VIC pobiera z tego obszaru bajt po bajcie, odnajduje w generatorze znaków odpowiadające każdemu z nich kształty właściwych liter, które są następnie wyświetlane na ekranie w postaci np. tekstu. Standardowo obszar pamięci ekranu tekstowego leży od adresu $0400 do $07E7, czyli od 1024 do 2023 dziesiętnie. Do zapisu znajdującego się na ekranie tekstu używa się tzw. kodów ekranowych, które zapisuje się bezpośrednio do wymienionego wyżej obszaru. Oto prosty program w BASIC-u, który pozwala na obejrzenie wszystkich standardowo zdefiniowanych znaków przez wpisanie ich kodów bezpośrednio na ekran. Warto zauważyć, że kody ekranowe liter różnią się od tzw. standardu ASCII.

```
5    REM WPISANIE KOLORU
10   FOR T=55296 TO 55551:POKE T,1:NEXT
15   REM WPISANIE ZNAKOW
20   FOR T=0 TO 255:POKE 1024=T,T:NEXT
```

Pamięć w C-64 została podzielona dla potrzeb VIC'a na cztery bloki po 16 kilobajtów, bo jest on w stanie w jednym momencie „widzieć” tylko 16 kB. Podziału dokonano w ten sposób:

od $0000 do $3FFF (0 do 16383)	- bank nr 3,
od $4000 do $7FFF (16384 do 32767)	- bank nr 2,
od $8000 do $BFFF (32768 do 49151)	- bank nr 1,
od $C000 do $FFFF (49152 do 65535)	- bank nr 0.

Nietrudno zauważyć, że banki ponumerowano w kolej ności malejącej (od 3 do 0). Zmiany „widzianego” przez VIC'a banku dokonuje się poprzez zmianę zawartości komórki o adresie $DD00 (56576 dziesiętnie). Jak to przeprowadzić ? Poniżej podano dwa sposoby – w assemblerze :

```
        LDA   $DD00
        AND   #$FC
        ORA   #$ numer banku (0-3)
        STA   $DD00
        RTS
```

i w BASIC'u :

```
10   POKE 56576,PEEK(56576)AND252 OR NR
```

..gdzie zmienna NR zawiera numer banku (3-0).

Wybór określonego banku oznacza, że zarówno ekran tekstowy jak i dane dla naszych znaków muszą się znajdować w wyznaczonym przez niego obszarze. Do zmiany położenia ekranu tekstowego i genereretora znaków w obrębie danego banku służy rejestr VIC'a o adresie $D018 (53272 dziesiętnie). Cztery młodsze bity (0-15) zawierają informacje o położeniu generatora, a cztery starsze o położeniu ekranu tekstowego. Odpowiednie wartości znajdziecie w poniższych tabelach (w nawiasach umieszczono wartości dziesiętne) :

Położenie ekranu tekstowego		wartości dla $D018	
$0000	(0000)	$00	(0)
$0400	(1024)	$10	(16)
$0800	(2048)	$20	(32)
$0C00	(3072)	$30	(48)
$1000	(4096)	$40	(64)
$1400	(5120)	$50	(80)
$1800	(6144)	$60	(96)
$1C00	(7168)	$70	(112)
$2000	(8192)	$80	(128)
$2400	(9216)	$90	(144)
$2800	(10240)	$A0	(160)
$2C00	(11264)	$B0	(176)
$3000	(12288)	$C0	(192)
$3400	(13312)	$D0	(208)
$3800	(14336)	$E0	(224)
$3C00	(15360)	$F0	(240)

Położenie generatora znaków		wartości dla $D018	
$0000	(0)	$00	(0)
$0800	(2048)	$02	(2)
$1000	(4096)	$04	(4)
$1800	(6144)	$06	(6)
$2000	(8192)	$08	(8)
$2800	(10240)	$0A	(10)
$3000	(12288)	$0C	(12)
$3800	(14336)	$0E	(14)

Aby wpisać właściwą wartość do $D018 należy pobrać z tabeli pierwszej wartość odpowiadającą wybranemu położeniu ekranu tekstowego, potem dodać do niej licz-bę z tabeli drugiej, która jest podana przy odpowiadającemu nam położeniu generatora znaków, a następnie całość wpisać do wspomnianego wyżej rejestru.

W przypadku, gdy zmieniliśmy położenie ekranu a chcemy pracować w BASIC'u, należało by jeszcze zmienić zawartość komórki $0288 (648 dziesiętnie), która zawiera starszy bajt adresu położenia ekranu na potrzeby instrukcji PRINT, kasowania ekranu itp. Jeżeli używamy banku o numerze niższym niż 3, a więc leżącego wyżej w pamięci, należy pamiętać, że VIC widzi tylko 16 kB i tak np. adres rzeczywisty $5800 to dla VIC'a $1800 (ważne w przypadku odczytu danych z tabeli)!

Na koniec jeszcze jedna uwaga : obszary od $1000 do $lFFF i od $9000 do $9FFF nie mogą być wykorzystywane dla generatora znaków i ekranu tekstowego, ponieważ są one sprzętowo używane przez generator znaków zawarty w ROM komputera (tzn. VIC widzi tam zawsze tylko dane z pamięci stałej).

Zamieszczony w dalszej części wydruk pamięci komputera zawiera napisany w języku maszynowym poręczny program do edycji własnych znaków. Należy go wpisać korzystając z dowolnego monitora, a po dokładnym sprawdzeniu zapisać na taśmę lub dysk za pomocą instrukcji : S„edytor”,1,0801,0C70 lub S„edytor”,8,... w przypadku dyskietki. Program pozwala tworzyć własną czcionkę w obszarze generatora umieszczonego pod adresem $2000 (8192).

Edytor uruchamiamy z poziomu BASIC'a dyrektywą RUN.

```
.:0800  00  0B  08  C6  07  9E  32  30
.:0808  36  31  00  00  00  A9  17  A0
.:0810  08  20  1E  AB  4C  38  08  0D
.:0818  0D  05  53  4B  4F  50  49  4F
.:0820  57  41  43  20  44  41  4E  45
.:0828  20  5A  20  52  4F  4D  27  55
.:0830  20  28  54  2F  4E  29  3F  00
.:0838  20  E4  FF  C9  54  F0  06  C9
.:0840  4E  D0  F5  F0  29  78  A9  33
.:0848  85  01  A9  00  85  FB  85  FD
.:0850  A9  D0  85  FC  A9  20  85  FE
.:0858  A2  08  A0  00  B1  FB  91  FD
.:0860  C8  D0  F9  E6  FC  E6  FE  CA
.:0868  D0  F2  A9  37  85  01  A2  00
.:0870  BD  8E  08  9D  00  40  BD  8E
.:0878  09  9D  00  41  BD  8E  0A  9D
.:0880  00  42  BD  8E  0B  9D  00  43
.:0888  E8  D0  E5  4C  00  40  A2  00
.:0890  8E  20  D0  8E  21  D0  A2  00
.:0898  BD  18  40  F0  06  20  D2  FF
.:08A0  E8  D0  F5  4C  29  40  8E  08
.:08A8  05  93  0D  0D  B0  C0  C0  C0
.:08B0  C0  C0  C0  C0  C0  AE  00  A0
.:08B8  08  A9  0D  20  D2  FF  A9  DD
.:08C0  20  D2  FF  A2  08  A9  20  20
.:08C8  D2  FF  CA  D0  FA  A9  DD  20
.:08D0  D2  FF  88  D0  E4  A9  0D  20
.:08D8  D2  FF  A9  AD  20  D2  FF  A2
.:08E0  08  A9  C0  20  D2  FF  CA  D0
```

.:08E8	FA	A9	BD	20	D2	FF	A2	00
.:08F0	8D	70	40	F0	06	20	D2	FF
.:08F8	E8	D0	F5	4C	F0	40	0D	20
.:0900	2B	2F	2D	20	2E	2E	2E	5A
.:0908	4D	49	41	4E	41	20	5A	4E
.:0910	41	4B	55	0D	20	43	52	53
.:0918	52	20	2E	2E	2E	53	54	45
.:0920	52	4F	57	41	4E	49	45	0D
.:0928	20	53	50	41	43	4A	41	20
.:0930	2E	2E	2E	5A	41	50	41	4C
.:0938	41	4E	49	45	2F	47	41	53
.:0940	5A	45	4E	49	45	20	42	49
.:0948	54	4F	57	0D	20	46	2D	31
.:0950	20	2E	2E	2E	53	41	56	45
.:0958	20	44	41	54	41	0D	20	52
.:0960	55	4E	2D	53	54	4F	50	20
.:0968	2E	2E	2E	50	4F	57	52	4F
.:0970	54	20	44	4F	20	42	41	53
.:0978	49	43	27	41	0D	00	A2	00
.:0980	8A	9D	D0	06	E8	D0	F9	8E
.:0988	0F	01	A9	1E	8D	3C	04	E8
.:0990	8E	3C	D8	20	0B	41	4C	1F
.:0998	41	AD	0F	01	38	E9	14	A2
.:09A0	00	9D	00	04	18	69	01	E8
.:09A8	E0	28	D0	F5	60	CA	A9	01
.:09B0	9D	00	D8	CA	10	FA	A9	80
.:09B8	8D	8A	02	78	A9	7F	8D	0D
.:09C0	DC	A2	01	8E	1A	D0	A9	1B
.:09C8	8D	11	D0	A9	C0	8D	12	D0
.:09D0	A9	50	8D	14	03	A9	41	8D
.:09D8	15	03	58	4C	96	41	A2	04
.:09E0	CA	D0	FD	A9	02	8D	20	D0
.:09E8	8D	21	D0	A2	0B	CA	D0	FD
.:09F0	A9	06	8D	20	D0	8D	21	D0
.:09F8	A9	18	8D	18	D0	A9	FB	CD
.:0A00	12	D0	D0	FB	A2	0B	CA	D0
.:0A08	FD	A9	02	8D	20	D0	A2	0B
.:0A10	CA	D0	FD	8E	20	D0	8E	21
.:0A18	D0	A9	15	8D	18	D0	EE	19
.:0A20	D0	4C	31	EA	A9	00	8D	10
.:0A28	01	8D	11	01	20	A4	41	4C
.:0A30	77	42	A9	00	85	FC	AD	0F
.:0A38	01	0A	26	FC	0A	26	FC	0A
.:0A40	26	FC	85	FB	A5	FC	18	69
.:0A48	20	85	FC	A0	07	B1	FB	99
.:0A50	40	03	88	10	F8	A9	79	85
.:0A58	20	A9	04	85	21	A2	00	A0
.:0A60	07	5E	40	03	90	03	A9	2A
.:0A68	FC	A9	2E	91	20	88	10	F1
.:0A70	A5	20	18	69	28	85	20	90
.:0A78	02	E6	21	E8	E0	08	D0	DF
.:0A80	A9	04	85	21	A9	79	85	20
.:0A88	AE	11	01	F0	0C	18	69	28
.:0A90	90	02	E6	21	CA	D0	F6	85
.:0A98	20	A5	20	18	6D	10	01	85
.:0AA0	20	90	02	E6	21	A0	00	B1
.:0AAB	20	09	80	91	20	A5	20	85
.:0AB0	FD	A5	21	85	FE	60	A9	00
.:0AB8	85	FC	AD	0F	01	0A	26	FC

.:0AC0	0A	26	FC	0A	26	FC	85	FB
.:0AC8	A5	FC	18	69	20	85	FC	A9
.:0AD0	79	85	20	A9	04	85	21	A2
.:0AD8	00	A0	07	BI	20	29	04	F0
.:0AE0	02	18	F4	38	7E	40	03	88
.:0AE8	10	F1	A5	20	18	69	28	85
.:0AF0	20	90	02	E6	21	E8	E0	08
.:0AFB	D0	DF	A0	07	B9	40	03	91
.:0B00	FB	88	10	FB	60	20	28	42
.:0808	20	A4	41	20	E4	FF	F0	FB
.:0B10	C9	03	D0	0C	78	20	A3	FD
.:0818	20	15	FD	20	18	E5	58	60
.:0B20	C9	2B	D0	0C	20	28	42	EE
.:0B28	0F	01	20	0B	41	4C	7A	42
.:0830	C9	2D	D0	09	20	28	42	CE
.:0B38	0F	01	4C	9C	42	C9	20	D0
.:0B40	0B	A0	00	B1	FD	49	04	91
.:0B48	FD	4C	FC	42	C9	11	D0	0E
.:0B50	AD	11	01	18	69	01	29	07
.:0B58	8D	11	01	4C	7A	42	C9	91
.:0B60	D0	09	AD	11	01	38	E9	01
.:0B68	4C	C8	42	C9	10	D0	0E	AD
.:0B70	10	01	18	69	01	29	07	8D
.:0B78	10	01	4C	7A	42	C9	9D	D0
.:0B80	0F	AD	10	01	38	E9	01	4C
.:0B88	E7	42	20	28	42	4C	EI	42
.:0B90	C9	93	D0	0F	20	28	42	A0
.:0B98	07	A9	00	91	FB	88	10	FB
.:0BA0	4C	7A	42	C9	85	F0	03	4C
.:0BA8	7D	42	20	86	42	A9	00	8D
.:0BB0	20	D0	8D	21	D0	A2	00	BD
.:0BB8	37	43	F0	06	20	D2	FF	E8
.:0BC0	D0	F5	4C	59	43	05	93	BE
.:0BC8	08	4E	55	4D	45	52	20	55
.:0BD0	52	5A	41	44	5A	45	4E	49
.:0BD8	41	20	28	31	20	4C	55	42
.:0BE0	20	38	29	3A	38	9D	00	20
.:0BE8	CF	FF	C9	31	F0	04	C9	38
.:0BF0	D0	C3	29	09	8D	0E	01	A2
.:0BF8	00	BD	79	43	F0	06	20	D2
.:0C00	FF	E8	D0	F5	4C	92	43	0D
.:0C08	0D	4E	41	5A	57	41	20	5A
.:0C10	42	49	4F	52	55	20	44	4C
.:0C18	41	20	53	41	56	45	3A	00
.:0C20	A2	00	20	CF	FF	C9	0D	F0
.:0C28	08	9D	40	03	E8	E0	10	D0
.:0C30	F1	8E	0D	01	A9	80	85	9D
.:0C38	A9	0D	20	D2	FF	20	D2	FF
.:0C40	AE	0E	01	A9	08	A8	20	BA
.:0C48	FF	AD	0D	01	A2	40	A0	03
.:0C50	20	BD	FF	A9	00	85	FB	A9
.:0C58	20	85	FC	A9	FB	A2	00	A0
.:0C60	28	20	D8	FF	4C	00	40	50
.:0C68	4F	4C	4F	4E	55	53	00	00

Listing programu „Edytor"

Paweł Sołtysiński

Biblioteki – część II

W dzisiejszym kąciku kodera będziemy kontynuować rozpoczęty miesiąc temu cykl na temat dos.library. Ostatnio omówiliśmy procedury Open, Close, Read i Write.

Rozpoczynamy od kolejnych procedur dos.library: Input (-54) i Output (-60). Służą one do komunikacji z oknem CLI (AmigaDOS). Do tych procedur wchodzimy bez żadnych wartości a w D0 zwracają nam one handle dla CLI. Jeżeli chcemy wysłać tekst do tego okna musimy użyć procedury Input a handle z D0 używamy przy procedurze Write , natomiast gdy chcemy odczytywać teksty wpisywane z klawiatury do tego okna to wpisujemy handle (z procedury Input) do D1, do D2 wpisujemy nasz bufor a do D3 długość tekstu, który chcemy wczytać i wykonujemy skok do procedury Read.

Teraz kolej na procedury odczytu katalogu. Do tego celu służą procedury Lock, UnLock, Examine i ExNext a także Info. Do procedury Lock (-84) wchodzimy z nazwą katalogu w D1 i trybem w D2. Zwraca nam ona lock naszego katalogu. Procedura Lock jest czymś w rodzaju procedury Open z tą różnicą, że odnosi się do katalogów. Procedura UnLock (-90) służy do zamknięcia katalogu otwartego przez Lock.

```
Procedura:   Lock
Adres:       -84/dos.library
Wejście:     D1 – nazwa
             D2 - tryb (ACCES_READ, ACCES_WRITE=-1
Wyjście:     D0 - lock katalogu
Błąd:        gdy D0 jest równe zero

Procedura:   UnLock
Adres:       -90/dos.library
Wejście:     D1 – lock
```

Gdy mamy otwarty (lock) dany katalog (np. DF0:) to możemy odczytac poszczególne pozycje katalogu za pomocą procedur Examine i ExNext. Procedura Examine odczytuje nazwę dysku a ExNext pozycje w katalogu.

```
Procedura:   Examine
Adres:       -102/dos.library
Wejście:     D1 - lock D2 - info buffer (bufor na
             dane dotyczące zbioru)
Wyjście:     D0 - status (zero oznacza pusty katalog)

Procedura:   ExNextAdres: - 108/dos.library
Wejście:     D1 - lock D2 - info buffer (bufor na
             dane dotyczące zbioru)
Wyjście:     D0 - status (zero oznacza koniec katalogu).
```

Nasz bufor (info buffer) na dane o zbiorze po wykonaniu procedury Examine lub ExNext będzie wyglądał następująco: (adres to adres względem początku bufora, przy nazwie .L oznacza, że wartością jest długie słowo):

Adres	Nazwa	Komentarz
0	DiskKey.L	
4	DirEntryType.L	Typ zbioru (ujemny = zbiór, dodatni = katalog)
8	FileName	Nazwa danych (108 bajtów)
116	Protection.L	Dana dotycząca protekcji zbioru (RWED itp.)
120	EntryType.L	
124	Size.L	Długość zbioru w bajtach
128	NumBlocks.L	Ilość bloków zajmowanych przez zbiór
132	Days.L	Data i czas utworzenia zbioru
136	Minute.L	
140	Tick.L	
144	Comment	Komentarz

Przykładowa procedura na odczytanie katalogu:

```
Lock:        = -84
UnLock:      = -90
Examine:     = -102
ExNext:      = -108
       ...
       move.l  dosbase,a6        ;baza dos.library
       move.l  #nazwa,d1         ;nazwa katalogu
       move.l  #-2,d2            ;odczyt
       jsr     Lock(a6)
       tst.l   d0                ;czy katalog istnieje?
       beq.s   error
       move.l  d0,lockzap
       move.l  lockzap,d1        ;lock do D1
       move.l  #bufor,d2         ;bufor na dane o zbiorze
       jsr     Examine(a6)
       tst.l   d0
       bne.s   jest
       bsr     print
pętla: move.l  zaplock,d1
       move.l  #bufor,d2
       jsr     ExNext(a6)
       tst.l   d0
       beq.s   end               ;czy koniec katalogu?
       bsr     print
       bra.s   pętla
end:   ...
print:         ; procedura np. druku nazwy
error:         ; procedura obsługi błędu
lockzap: dc.l  0
bufor: blk.l   260
nazwa:dc.b     'DF0:',0
```

Następnymi procedurami są: CurrentDir i CreateDir. Procedura CurrentDir służy do uaktualnienia danego katalogu (działanie jak CD z Amiga DOS'u) natomiast CreateDir umożliwia nam utworzenie danego katalogu.

```
Procedura:   CreateDir
Adres:       -120/dos.library
Wejście:     D1 – nazwa
Wyjście:     D0 - lock utworzonego katalogu (jeżeli równy
             zero to błąd)

Procedura:   CurrentDir
Adres:       -126/dos.library
Wejście:     D1 – lock
Wyjście:     D0 - poprzedni lock.
```

```
CreateDir:     = -120
       ...
       move.l  dosbase,a6        ;baza dos.library
       move.l  #nazwa,d1         ;nazwa katalogu
       jsr     CreateDir(a6)
       tst.l   d0                ;czy katalog utworzony?
       beq.s   error
       ...
error:         ; procedura obsługi błędu
nazwa: dc.b    'DF0:',0.
```

Przykładowa procedura na utworzenie katalogu:

Ostatnimi omawianymi dziś procedurami będą: LoadSeg i UnLoadSeg. Służą one do wczytywania zbiorów (object) wraz z zaalokowaniem pamięci na nie i ich relokacją.

```
Procedura:   LoadSeg
Adres:       -150/dos.library
Wejście:     D1 - nazwa zbioru
Wyjście:     D0 - segment (adres zbioru = segment * 4).

Procedura:   UnLoadSeq
Adres:       -156/dos.hbrary
Wejście:     D1 – segment.
```

Życzę wielu owocnych godzin przy klawiaturze tym wszystkim, którzy mają trochę większe ambicje niż przełączanie opcji w X-Copy.

Duddie

Zmagania z Copper'em!

AGNUS to procesor graficzny o najwyższym priorytecie dostępu, co oznacza, że nie może go zatrzymać żaden z koprocesorów Amigi (włącznie z głównym procesorem MC68000).

W tej kości mieszczą się dwa koprocesory – Copper i Blitter. Copper jest procesorem obrazu (tworzenie i wyświetlanie) a Blitter to procesor graficzny, który może wykonywać operacje logiczne na obszarach pamięci. Zarówno Blitter jak i Copper mogą pracować jedynie na pamięci typu Chip i w zależności od typu Copper'a może on zaadresować 512kB lub 1024kB pamięci. Agnus, który pracuje na 1MB to tak zwany Big Fat Agnus i montowany jest on w nowych modelach Amigi (Amiga 500 z nowym Agnusem ma miejsce na płycie na dodatkowe 0.5MB Chip-Ram'u). Nowy Agnus ma rozszerzone możliwości, ale o tym później.

Wszystkie układy, do których dostęp odbywa się przez kanały DMA posiadają swoje rejestry, które znajdują się od adresu $dff000 (tzw. custom). Znajdziemy tam adresy bitplane'ów, dane o kolorach, tryb wyświetlania, itp. Jednak najczęściej nie możemy sprawdzić zawartości tych komórek, ponieważ w większości są to komórki tylko zapisywalne. Używając ich można osiągnąć bardzo ciekawe efekty. Zajmijmy się na przykład wyświetlaniem obrazu. Każdy obraz składa się z bitplane'ów. Ilość włączonych bitplane'ów decyduje o ilości kolorów na ekranie. I tak dla 1 bitplane'u mamy 2 kolory (1 kolor tła i 1 atramentu). Ilość kolorów to 2 do potęgi równej ilości włączonych bitplane'ów. Zapalone lub zgaszone bity na tych bitplane'ach decydują o numerze koloru dla danego punktu ekranu. Bitplane pierwszy decyduje o bicie zerowym, bitplane drugi o bicie pierwszym, itd. Kolory umieszczamy w komórkach obsługiwanych przez Copper (kolor jest definiowany przez słowo -2 bajty) pamięci:

dla koloru 1 - $dff180 - COLOR00,
dla koloru 2 - $dff182 – COLOR01,
dla koloru 32 - $dff1be – COLOR31.

Adresy bitplane'ów umieszczamy natomiast w komórkach (definiowane przez długie słowo - 4 bajty):

Bitplane 1 - $dff0e0 – BPL1PT,
Bitplane 2 - $dff0e4 - BPL2PT ...
Bitplane 6 -$dff0f4 – BPL6PT.

Jak widać mamy 32 kolory definiowane i 6 bitplane'ów.

O ilości bitplane'ów oraz o trybie wyświetlania decyduje komórka: BPLCONO0-$dff100. Poszczególne bity tej komórki określają funkcje jakie ma wykonywać procesor odpowiadający za wyświetlanie obrazu, i tak: Bit 15 - HIRES - oznacza tryb wysokiej rozdzielczości. Bit 14 - BPU2, Bit 13 - BPU 1, Bit 12 - BPU0. Bity BPU wyznaczają ilość użytych bitplane'ów. Bit 11 - HOMOD - włączony tryb HAM (Hold And Modify), Bit 10- DBLPF - włączony tryb Double Playfield, w którym dysponujemy dwoma niezależnymi ekranami o liczbie kolorów nie przekraczającej 8, Bit 9 - COLOR - kolory na wyjściu Composite Video, Bit 8- GAUD - wykorzystywany w przy przypadku interfejsu Genlock, Bit 3 - LPEN - używany w przypadku pióra świetlnego, Bit 2 - LACE - tryb Interlace (wybieranie międzyliniowe), Bit 1- ERSY - synchronizacja zewnętrzna. Bity 7,6,5,4,0 - niewykorzystane. Jeżeli więc chcemy obraz 16 kolorowy w Hi-Res'ie to musimy wpisać do BPLCON0 wartość $c000 (ustawiony bit HIRES oraz włączone 4 bitplane'y).

Kolejnymi komórkami związanymi z bitplane'ami są: BPLCON 1-$dff102, i BPLCON2 -$dff104. BPLCON1 odpowiada za przesunięcie w lewo (w granicach od 0 do 15 pixel'i) bitplane'ów. Bity 0-3 odpowiadają wartości przesunięcia plane'ów parzystych a bity 4-7 plane'ów nieparzystych. BPLCON2 decyduje o priorytetach sprite'ów lub ekranów (jeżeli jesteśmy w trybie Double Playfield). Bit 6 - plane'y nieparzyste mają priorytet nad parzystymi (tylko w trybie Double Playfield). Bity 5-3 - priorytety plane'ów nieparzystych. Bity 2-0 - priorytety plane',ów parzystych.

Aby umożliwić lepsze wykorzystanie procesora Copper w Amidze ma swój własny program, który jest wykonywany niezależnie od programu wykonywanego przez procesor. CopperList (czyli program dla Coppera) musi znajdować się w pamięci Chip a jego adres wpisujemy do komorki: COP1LC -$dff080. Jego uruchomienie następuje przez wpisanie 0 do komórki: COPJMP1 -$dff088. W celu ułatwienia pracy (np. w trybie Interlace) Amiga posiada możliwość zdefiniowania drugiego CopperList'a. Gdy mamy zdefiniowane dwa CopperList'y nie musimy dbać o zmianę adresu w jednej komórce ale po prostu na zmianę uruchamiamy to pierwszy to drugi CL. A adresy dla drugiego CL to: COP2LC - $dff084 COPJMP2 - $dff08a.

O pisaniu programów dla Coppera napiszemy w kolejnych odcinkach.

Marcin „Duddie” Dudar

GRACZ DOSKONAŁY czyli jak oszukiwać

Rubryka ta jest dla tych wszystkich, którzy lubią ułatwiać sobie granie. Na początek pozwolę sobie zacytować słowa, które znalazłem w swoich notatkach: „No... gdyby udało mi się jeszcze znaleźć sposób by uzyskać nieskończoną ilość żyć w tej największej grze jaką jest życie, albo chociaż edytor". A teraz zabieramy się za gry komputerowe.

After The War

By stać sic całkowicie nieśmiertelnym należy przycisnąć jednocześnie następujące klawisze: ALT, 1, B – w pierwszej części ALT, 1, M - w drugiej części. Niecierpliwym podpowiem, że hasłem do drugiej części jest: 101069.

Hybris

Wgraj grę i poczekaj, aż pojawi się tabela najlepszych wyników. Teraz wpisz „COMMANDER" i wciśnij fire. Po chwili wciśnij F10. Jeśli twój statek mrugnie to oznacza, że jesteś nieśmiertelny i posiadasz nieskończoną ilość rozszerzeń i bomb. Wciskając: F1-F5 = wybierasz rodzaj rozszerzenia, F8 = stopień trudności (są trzy), F10 = włącza i wyłącza ułatwienia.

Strider

Wciśnij F9 by zatrzymać grę. Teraz wciśnij jednocześnie lewy Shift, HELP i 1. Następnie wciśnij F10 by zacząć grę od początku. Wciskając: 1-5 = przenosisz się do odpowiednich stref, F1-F4 = przenosisz się do odpowiednich sekcji w danej strefie oraz odnawiana jest energia.

Populos

Znajomość haseł odpowiednich poziomów pozwoli Ci na sprawdzenie swoich sił z różnymi przeciwnikami. Oto niektóre z nich:

POZIOM	HASŁO
225	HURTOGODOR
250	VERYOXT
275	BUGINOND
300	BILQUAZOUT
325	SCODEING
350	SUYDIEHOLE
375	SADOUTER
400	BADMEILL
425	BURIKEPIL
450	JOSYMAR
475	MINCEME
500	SHADWILDON
999	KILLUSPAL

Platoon

Wpisz „HAMBURGER-HILL" używając znaku minus na klawiaturze numerycznej. Teraz wciskając F1-F4 przenosisz się do odpowiednich stref.

SimCity

Trzymając SHIFT wpisz „FUND" by uzyskać dodatkowe $10.000. Trzykrotne powtórzenie tej operacji powoduje trzęsienie ziemi, więc zbierz fundusze zanim rozpoczniesz budowę miasta. By oszukać mieszkańców ustaw podatki na 0%. W grudniu (lub w listopadzie jeśli grasz z fast speed) zmień je na 20%. Po zebraniu podatków ponownie ustaw je na 0%. W ten sposób mieszkańcy myślą, że nie płacą podatków.

Stormlord

W czasie gdy gra się ładuje wciśnij fire oraz lewy przycisk myszy. Trzymaj je dopóki gra nie skończy się ładować. Gdy gra się rozpocznie zatrzymaj ją wciskając SPACE. Teraz wpisz „MNBVC". Nagrodą jest nieskoficzona ilość żyć oraz czasu. Wciskając „L" zmieniasz strefy.

Supercars

By rozpocząć grę z $500.000 wpisz jako imię „RICH".

Untouchables

Zanim rozpoczniesz grę wpisz „SOUTHAMPTON-GAZETTE". Jeśli teraz wciśniesz F10 to przeniesiesz się do kolejnego poziomu. Na poziomie 2,3 i 6 wciskając HELP przenosisz się do kolejnej strefy poziomu.

X - Out

W sklepie wybierz statek o kształcie białej pluskwy. Następnie wybierz pojedyńczy pomarańczowy laser i przystaw go do twarzy właściciela sklepu. Teraz wciskaj lewy przycisk myszy do czasu aż otrzymasz 500.000 kredytów.

Mr. Raf

OPERATION STEALTH – całkiem serio o wojnie

No i nareszcie jest. Po dużych trudach dotarła do nas dobra wersja gry „Secret Defense Operation Stealth ". Po pierwszym zauroczeniu muszę, niestety, stwierdzić, że tym razem „Delphine Software" nie dopracowało swojego produktu. Często zdarza się, że gra „pada" (np.: ponownie prosi o włożenie pierwszego dysku, gdy ten nie był wyciągany ze stacji). Oprócz błędów programowych autorzy przesadzili z ilością typowo zręcznościowych części. Nie tego spodziewali się miłośnicy gier przygodowych. Gra była jednak na tyle ciekawa by wciągnąć nas (Himan'a i mnie) na całe dwa dni, bowiem tyle czasu zajęło nam jej ukończenie. Sądzę, że jesteście gotowi wkroczyć do akcji i po raz kolejny uratować Świat.

Mr. John Glames - najlepszy agent C.I.A. (taki amerykański James Bond) to właśnie Ty. Kolejny raz nie pozwolono Ci spokojnie spędzić urlopu, ale sytuacja jest poważna.

Z bazy Miramar skradziono najnowszy, niewidoczny dla radaru myśliwiec Stealth. Brak dowodów uniemożliwia całkowite zidentyfikowanie sprawcy, lecz nie zrobili tego Rosjanie. Przypuszczalnie samolot wylądował w bananowej republice Santa Paragua. Tam właśnie się udajesz.

Spragniony wieści ze świata podchodzisz do automatu z gazetami. Moneta do odpowiedniej szczeliny i wyciągasz gazetę, z którą udajesz się do ubikacji. Z niej dowiadujesz się, z którym państwem nawiązano dobre stosunki. Otwierasz walizkę i zabierasz pióro oraz pieniądze. W tajnym schowku za Pomocą urządzenia sporządzasz fałszywy paszport. Bez problemu wkraczasz na teren państwa. Hostessa wręcza Ci telegram, z którego wiesz jaki bagaż masz zabrać. Pokazujesz bilet i z torbą pana Martineza wchodzisz do toalety. Zabierasz zegarek i maszynkę do golenia, a sama torba ląduje w śmieciach. Maszynka to magnetofon, więc włączasz go do gniazdka i otrzymujesz instrukcje, gdzie masz się spotkać z agentem. Taksówka dowozi Cię do śródmieścia. By nie wzbudzać podejrzeń udajesz się do banku i rozmieniasz tam wszystkie pieniądze. Teraz zgodnie z instrukcją kupujesz u kwiaciarki kwiatek i wpinasz go w butonierkę. Następnie udajesz się do parku gdzie na ławce oczekujesz na agenta. Sytuacja zaczyna się komplikować. Agent zostaje postrzelony z przejeżdżającego samochodu. Ostatnim wysiłkiem przekazuje Ci kartkę z kluczem. Ponownie udajesz się do banku, gdzie kartkę pokazujesz urzędnikowi (najpierw oderwij klucz), który otworzy Ci pomieszczenie ze skrytkami. Kluczem otwórz odpowiednią skrytkę. W niej znajdziesz walizkę z podwójnym dnem. Zabierz elektroniczny otwieracz sejfów i kopertę. I tu po raz pierwszy spotykasz się z dwoma rosyjskimi szpiegami: pułkownikiem Karpov'em i towarzyszem Ostrovitch'em. Koperta z tajną dokumentacją wędruje do ich kieszeni, a oni prowadzą Cię do opuszczonej kopalni. Dokładnie skrępowanego zasypują Cię za pomocą dokładnie umieszczonego ładunku dynamitu. Pod palcami wyczuwasz kawałek metalu, o który przecinasz swoje więzy i odkopujesz górniczy kilof porzucony dawno temu. Badając ściany odkrywasz powiew świeżego powietrza i tę ścianę kujesz odsłaniając przejście.

Tu mamy pierwszą zręcznościówkę. Kierując naszym bohaterem musisz przebyć podwodne groty. Dla ułatwienia powiem: w ostatniej fazie zrób sobie odpoczynek w połowie ekranu. Znowu jesteś w mieście. Udaj się do hotelu, lecz najpierw zakup na plaży specjalną bransoletkę. W hotelu wjedź na drugie piętro i schodami wejdź na trzecie. Otwórz ostatnie drzwi w rzędzie i wejdź do środka. Tu po raz kolejny sytuacja zaczyna się komplikować. Najpierw trzyma Cię na muszce piękna młoda kobieta i oskarża o bestialstwo, a za chwilę ratuje (?) Cię twój sobowtór. Sobowtór to bezwzględny morderca Otto, a piękna kobieta to Miss Julia Manigua, siostrzenica prezydenta.

Jako miejsce docelowe morskiej podróży wybrano okoliczną głębinę, w której spoczną Wasze ciała odpowiednio obciążone kamieniami. Na statku uruchom bransoletkę i spokojnie pozwól by wrzucono Cię

do wody. Na dnie uwolnij się z więzów i oswobodź dziewczynę. Dzięki pomocy szefa partyzantów Tania dostaniecie się do pałacu. Tu Julia zostaje porwana, a Ty wkraczasz w kompleks labiryntów, które strzegą pokoju prezydenta. W pokoju naciśnij ramię figurki by odsłonić sejf. Teraz użyj elektronicznego otwieracza sejfów (przy każdej cyfrze zapali się światełko pokazujące jej miejsce w kombinacji) i zabierz kopertę z planami. Ponownie spotykasz się z Rosjanami, by po śmiertelnej jeździe skuterem wodnym odebrać swoją własność i dotrzeć do łodzi podwodnej. Tutaj otrzymujesz wyjaśnienia co do dalszego postępowania oraz uzupełniasz swój zapas gadget'ów fabrykowanych przez niezastąpionego Charlie'go. Wydostajesz się z łodzi przez luk torpedowy i rozpoczynasz przeszukiwanie dna morskiego uważając na rekiny. Tam znajdziesz elastyczną taśmę. W pniu jednej z zatopionych palm znajduje się przycisk, który otwiera tajne przejście. Znów wpadasz w pułapkę zastawioną przez diabolicznego Dr. Who, który jest szefem organizacji Spider. Zamierzają oni dzięki porwanemu bombowcowi zniszczyć wszystkie stolice świata. Ciebie natomiast umieszczają w wolno opuszczającej się klatce, która wisi nad basenem z piraniami. Gdy tylko zostaniesz sam za pomocą pióra otwórz zamek klatki. Zegarek dostarczy Ci linę, po której dostaniesz się do przewodów wentylacyjnych. Tu kolejny labirynt na końcu którego oczekuje kolejna porcja przygód. Wychodzisz w łazience, gdzie jeden z żołnierzy dokonuje codziennej toalety. Bez słowa sprawdzasz na nim twe umiejętności w walce wręcz, kneblujesz go ręcznikiem i uniemożliwiasz ruch sznurówkami.

Szybka zmiana ubrania i spokojnie wkraczasz do magazynu. Tu z szuflad zabierasz sznurówki (szybko ich użyj) oraz drewniany stempel. Oficer w korytarzu wysyła cię po szklankę wody. Znalezioną w łazience szklankę napełniasz wodą z zasobnika znajdującego się w kajucie.

Teraz musisz działać szybko. W pokoju dowodzenia podajesz oficerowi szklankę i wykorzystując chwilę jego nieuwagi podmieniasz stemple. Ze szklanką udajesz się do toalety, gdzie za pomocą bibułki z papierosa (są w papierośnicy) o kolorze niebieskim sporządzasz fałszywe odciski palców. Ponownie udajesz się do kajuty, gdzie zabierasz poduszkę do stempli, rozkaz na piśmie oraz łódź ratunkową. Teraz podstempluj rozkaz i udaj się do drzwi z zamkiem papilarnym. Użyj fałszywych odcisków i wejdź do środka. Przez szczelinę w ścianie podaj rozkaz strażnikowi, który otworzy wejście. W korytarzu podłącz elektryczną golarkę do kontaktu i włącz ją. Teraz umieść ją w koszu i wejdź do centrum sterowania. Tu ponownie okazuje się, że wszyscy przygotowani są na Twoje pojawienie się. Jesteś świadkiem startu bombowca, który niesie atomową śmierć. W pewnym momencie za ścianą odzywa się pozostawiony przez ciebie magnetofon. To odwraca uwagę Otta i pozwala Ci odpalić rakietkę w komputer (rakieta to papieros z czerwoną obwódką). Nie zwracaj uwagi na gadanie doktora tylko jak najszybciej potraktuj Otta tak jak wcześniej żołnierza w toalecie. Dr. Who ponownie porywa Julię, lecz nie jest to w tym momencie najważniesze.

Teraz jak najszybciej musisz wprowadzić do czytnika compact, na którym znajduje się wirus. To spowoduje, że Stealth Fighter stanie się „widoczny" dla radaru i będzie go można zniszczyć. Gdy już to uczynisz uciekaj z centrum dowodzenia , gdyż za parę minut cała wyspa przestanie istnieć.Dr. Who postanawia pozbyć się zakładniczki pozostawiając ją ogłuszoną na płycie lotniska. Szybko ją cucisz i razem chwytacie się płozy helikoptera by nim opuścić niebezpieczne miejsce. Wyspa ponownie znika w głębinach, ale tym razem na zawsze. To jednak nie koniec. Pod helikopterem podwieszona jest bomba, którą na wszelki wypadek „potraktuj" elastyczną taśmą. Teraz w locie na spotkanie morza wyrzuć tratwę ratunkową. Spokojnie obserwujesz jak bomba, która miała was zlikwidować uderza, dzięki działaniu taśmy, w helikopter. To już w zasadzie wszystko.

Pozostaje jeszcze tylko końcowa scena, którą wytrwali obejrzą z przyjemnością.

Mr. Raf

Public Domain

Przypuszczamy, że wielu czytelników zetknęło się już z tym terminem. Wyjaśnijmy jednak, że określenie Public Domain można przetłumaczyć jako „własność publiczna". Nazwę tę nadano dużej grupie programów, które nie są czyjąś konkretną własnością, a więc wszyscy użytkownicy mogą je posiadać i z nich korzystać bez obawy o ich nielegalne pochodzenie itp.

Najczęściej spotykanymi przykładami „Public Domain" są programy demonstracyjne (tzw. dema) i programy użytkowe, które zostały przez autorów oddane w ręce użytkowników na zasadzie własności publicznej.

Wzorem renomowanych magazynów komputerowych również i my postanowiliśmy umożliwić czytelnikom „64 plus 4" dostęp do najświeższych pozycji Puplic Domain z kraju i ze świata. Tak więc co miesiąc będzie można otrzymać w naszej redakcji **„64 plus 4 PUBLIC DOMAIN PACK"!**

Począwszy od stycznia czeka na Was zestaw świeżych programów demonstracyjnych i użytkowych dla C-64 i AMIGI. Oprócz programów Public Domain na dyskietkach zamieszczać będziemy programy, których listingi prezentujemy w naszym piśmie oraz ciekawsze prace nadesłane na nasze konkursy. W zestawach umieszczać będziemy również nadesłane nam programy, które z racji np. swej długości (m. in. własne programy demonstracyjne) nie będą mogły startować w naszym konkursie na najlepszy program miesiąca.

Dyskietki z Public Domain będą przygotowywane tylko dla komputerów C-64 i AMIGA. Rodzina C-16, jak na razie, nie może poszczycić się regularnym dopływem odpowiedniej ilości oprogramowania. Oczywiście nie mamy tu na myśli gier, które najczęściej własnością publiczną nie są.

Zestawy „64 plus 4 PUBLIC DOMAIN PACK" należy zamawiać drogą pocztową wpłacając na konto bankowe redakcji kwoty:

- **20.000 zł za „64 plus 4 PUBLIC DOMAIN PACK – C-64",**
- **25.000 zł za „64 plus 4 PUBLIC DOMAIN PACK – A".**

Kwoty te obejmują koszt dyskietki (dla C-64 dysk 5.25", dla Amigi 3.5"), oraz koszty kopiowania, opakowania i przesyłki pocztowej. Chcąc zapewnić sobie stałe otrzymywanie naszych „Pack'ów" można wykupić prenumeratę. Przyjmujemy ją na podstawie wpłaty dokonanej na konto:

Bank PKO SA Oddział w Bydgoszczy,
konto nr: 5.09011-400522.7-136-11-111.0.

Blankiety wpłat powinny być CZYTELNIE wypełnione i zawierać: imię i nazwisko (lub nazwę instytucji), dokładny adres zamawiającego, skrót „PDP-64" (jeśli zamawiamy zestaw dla C-64) lub „PDP-A" (jeśli zamawiamy zestaw dla Amigi). Cena jednego egzemplarza PDP-64 w prenumeracie wynosi 18.000 zł, (12 numerów - 216 tys. zł) a PDP-A 22.000 zł (12 numerów - 264 tys. zł). Prenumeratę można zawrzeć w dowolnym miesiącu na okres od trzech do dwunastu miesięcy (do końca roku kalendarzowego). Wśród pierwszych stu prenumeratorów rozlosowane będą atrakcyjne nagrody! Planujemy również wielkie głosowanie (i wielkie nagrody) na najlepsze demo z naszych „Pack'ów".

NAUKA I ŚWIETNA ZABAWA,
CO MIESIĄC NOWOŚCI
- TO WSZYSTKO ZAPEWNI CI NASZ
„64 PLUS 4
PUBLIC DOMAIN PACK" !!

Redakcja

P.S. Spis zawartości styczniowych zestawów na stronie 11.